Github stuff

http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1

Tutorial:

1. What is Github and Git? The way I describe it as is a sort of Cloud, for free we are given unlimited space for free at the cost all of our data is open to the public. But for a small cost a month we can chose to only open it to specific people. In actually there is more to Github, such as what truly makes it amazing is the fact that it saves the past states of our codes and other documents allowing us to see changes made over time and allows people to make different changes to a document and save them as two seperate versions without them being overwritten.

2. How do we get Github and Git? Github the online version of Git can be found at: https://github.com/ this website requires that we also get Git, a group of command line functions that are found at: http://git-scm.com/downloads .

3. Once you have Git downloaded we can create an account in Github, from there we can create an online repository that is ready to use. This requires no command line or terminal action to get going.

4. Now this maybe okay, but we want to fully utilize the powerful language Git, so the further steps are to help you understand what Git can do on the Command line or Terminal.

5. To first setup Git and creating local repositories on your computer is to type in a folder :

```
git init
```

6. Now let us write a .txt file in our local git repository

7. Once we are done writing the .txt file let us use the command:

```
git status
```

This will command will show our .txt file is “untracked”.

8. Now let us use the command:

```
git add *.txt
```

This command will basically add the specified file into a “queue” until our next commit.

9. Now that we have added our file to the queue will now track the queue using

```
git commit
```

We can then check git status and see the our file is being tracked.

10. So far we have been only working in our own local git repository so let us look at transferring files from our local repository to our online GitHub repository.

11. We will use the git add command in a new way:

```
git remote add origin https://github.com/githubusername/repositoryname
```

This command queue our repository in order to be pushed to our online repository.

12. Now we use this command:

```
git push origin master
```

This command is like the commit command used in the local repositroy.

Read:

```
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master
```